Attorney Docket No. 08005.0009
Customer Number 22,852

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Ralf VIERICH et al.	)	Group Art Unit: 3623
	)	
Application No.: 10/624,489	)	Examiner:
	)	
Filed: July 23, 2003	)	
	)	
For: STATIC DRILL-THROUGH MODELLING	)	
	)	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

CLAIM FOR PRIORITY

Under the provisions of 35 U.S.C. § 119, Applicants hereby claim the benefit of the filing date of Canadian Patent Application No. 2,394,713, filed July 23, 2002, for the above-identified U.S. patent application.

In support of this claim for priority, enclosed is one certified copy of the priority application.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.

Dated: April 14, 2004

By: _____
Ernest F. Chapman
Reg. No. 25,961

EFC/FPD/gah
Enclosures

Office de la propriété intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An Agency of Industry Canada

Bureau canadien des brevets

Certification

Canadian Patent Office

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,394,713**, on July 23, 2002, by **COGNOS INCORPORATED,** assignee of Charles Michael Potter and Ralf Vierich, for "Static Drill-Through Modelling with Graphical Interface".

Sylvie Grégoire
Agent certificateur/Certifying Officer

August 21, 2003
Date

Canada

(CIPO 68)
04-09-02

OPIC CIPO

Abstract of the Disclosure

The invention, when incorporated in a drill-through modeling tool (DMT), allows all the elements affecting drill-through behavior to be aggregated at a single point, thereby allowing administration to be simplified, and also permitting easier understanding and integration with third-party tools. The invention also provides for graphical displays of drill-through paths for a DMT user or modeller. These displays show the parameters and dependencies of each drill-through path and allow users to obtain a quick overview of the drill-through network and further, they allow the tool user or modeller to confirm drill-through dependencies at a glance. Drill-through objects may thus be manipulated and maintained in a graphical manner.

Static Drill-through Modelling with a Graphical Interface

Field of the Invention

The present invention relates to a method of interactively searching a database in such a manner that it is quick and easy. It is particularly directed to mechanisms to drill-down, drill-up and drill across data collections to allow applications to present the user with summary information. The present invention also relates to retrieving information from a database based on content aggregation, management and distribution.

Background of the Invention

A key to success in business today is to understand and effectively manage the factors that drive an enterprise. Having critical information about business drivers allows decisions that will significantly improve results. As organizations 'flatten,' these mission-critical decisions are being made at lower levels—which means that almost every employee in an enterprise needs quick, easy access to appropriate information. In spite of this necessity, corporate information remains inaccessible for many employees — virtually locked away in data warehouses, data marts, enterprise resource planning systems, and a myriad of corporate databases.

Traditionally, the only way to retrieve valuable information from these data stores has been to rely on Information Technology (IT) specialists to build SQL-type queries that deliver transaction-level detail. The advent of On-Line Analytical Processing (OLAP), however, has given organizations more effective and meaningful access to critical corporate data. OLAP consolidates and presents summarized corporate information from a multitude of sources. This technology allows users to view information in a business context: sales per quarter per sales rep, units shipped on time per city per branch by air, and so on. By presenting corporate data in this way, trends and anomalies can be easily spotted and addressed.

OLAP reports - interactive reports that are highly formatted, easily deployed, and effortless to use - deliver value to the entire organization. These

reports accelerate the "Eureka moment" by exposing sweet spots of information in a data set directly to decision makers, knowledge workers, and information consumers. Sweet spots are selective chunks of information that provide decision makers with immediate critical insight into business drivers. These OLAP reports can be regular status reports, but are especially effective for key performance indicator (KPI) reporting, business performance measurement reporting, and scorecard reporting, all of which are becoming increasingly important to decision makers, who require a robust reporting environment in which to perform these tasks.

Products such as PowerPlay™ by Cognos Inc. enable organizations to create and deploy highly formatted, interactive OLAP reports. These reports let users easily measure, manage, and improve business performance, then distribute this information across the enterprise. Decision makers throughout the organization now have the information they need to significantly improve business results.

Because of the complexity of corporate organizations, and their data, it is not unusual for a particular enterprise to deploy several different products to analyze their data. Often these products are from different vendors.

For companies that want to track performance and trends, or perform score-carding-style management reporting, there is an even more fundamental concern. It can be extremely difficult to understand "the big picture" when the only accessible reports focus on transaction-level detail, because data in databases is organized for efficient storage and administration — not for summary-level analysis or exploration. In addition, data storage does not correspond to how the business is organized, so data must be massaged before the average user can extract useful information from it. If, for example, managers want to explore company performance in terms of product sales, a report that details the performance of individual sale reps will not help them spot overall trends. By reviewing summary information first, such as total sales per office or region, decision makers can more easily gain a "big picture" view of business

performance. They can then drill down to lower-level details to uncover what is driving these trends. Thus OLAP technology has brought significant value to business decision making. OLAP systems store and access data as dimensions that represent business factors like time, products, geographical regions, and sales channels. This information is stored "multi-dimensionally" - like a cube that can be viewed, turned, and explored from any angle. It is also presented in a business context, like 'number of customer complaints by product line in North America last quarter,' rather than a database context - so decision makers have immediate access to the information they need to make the best decisions for the business.

Until recently, organizations have found it difficult to meet some of these user requirements. However, with the advent of OLAP tools enterprise-wide deployment of OLAP reports is now a reality. Cubes can be customized to reflect the information (also called dimensions) and calculations (also called measures) most commonly used in a given organization. OLAP reports are generated from data cubes. Because each cube contains a wide variety of dimensions and measures, a vast number of reports can be built from the information in the cube. The cube can be considered as a master report or a collection of components that can be assembled to create a specific report.

With OLAP reports, the user's first view of the data is a 'top-level' one that reveals patterns and trends at a glance. If users have identified issues in this summary-level information, OLAP reports enable them to fully explore and analyze the data set from any angle, to any level of detail. They also enable users to 'slice and dice', drill down, and change graphical views of their data — something paper reports cannot offer.

OLAP reports that take an analyze-then-query approach allow decision makers to access data the same way they identify and solve problems: by reviewing totals or summary information first, then looking at the underlying details by drilling down to transaction-level details whenever necessary.

There are two stages to implementing a reporting solution. The first step is to create OLAP cubes, the multidimensional structures that house summary-level details of the corporate data. Typically, these cubes are created by IT specialists and deployed to information analysts and report authors. The cubes are customized models of a business that reflect the unique characteristics of the company. The structure of a cube is defined in terms of dimensions and measures. Dimensions are hierarchical categories of information like time, products, and geography. For example, the product dimension hierarchy may be organized by product line, product group, and then by individual product. Measures are the calculations that are used to track the business such as revenue, units sold, and cost of sales.

OLAP cubes generally contain only the dimensions and measures relevant to a specific analysis. For example, sales analysis data and human resources data would be housed in separate cubes. This ensures that cubes remain manageable, not just in terms of their size but also in terms of the clarity of the information they contain. With appropriate tools, cubes can be easily linked together so that users can move effortlessly from one cube to another, accessing information from all areas of the company.

Once OLAP cubes are created and deployed, report authors have everything they need to produce a wealth of OLAP reports. The process for authoring is extremely straightforward for all types of reports: status reports that reveal a snapshot of data; ad hoc reports that answer specific questions; and business performance management reports that track KPIs (Key Performance Indicators).

Although OLAP reports can be distributed on paper, it is well known that decision makers reap the most value when the reports are presented electronically. There are three ways to explore data in an OLAP report:

- Drill down/Drill up: Users can explore a dimension hierarchically -moving from summary-level information to the details and back - to gain fast answers to critical business questions. A financial manager who is concerned with rising

expenses will want to understand what parts of the company are particularly problematic. By drilling down on a geographical dimension, they can move from looking at expenses by country, by region, by office and then finally by department. Drilling up is the reverse process, so that the information becomes more summarized, and less detailed.

• Drill-through and Slice and dice: Decision makers can interactively explore corporate data in any combination of dimensions, from many different angles. For example, a sales manager can look at revenue figures by product line, sales region, time period, or sales channel.

• Graphical analysis: Users can choose from a variety of graphical displays with which to depict the key factors that are driving the business and assist them in understanding the performance of various aspects of the business.

In earlier products the mapping of source and target drill-through objects was spread over several report generating applications, sometimes obtained from different vendors, and used data also provided by different products, sometimes from different vendors. This made administration difficult since changing drill-through behavior required the administrator to apply the changes to more than one report generating application. It also meant that it was very difficult to deploy or assess dependencies for drill-through installations.

Another problem was that drill-through definitions were either saved in cubes or saved in report definitions, so that when a change was made, such as a report being deleted or modified, it was very hard for the administrator to determine what cubes or other reports were dependent on the report that was being deleted or modified (i.e. were possible drill-through targets).

Further, drill-through objects were somewhat "closed in" or concealed, that is, unintentionally, but effectively, hidden from external applications, so that it was difficult for third party applications to fully utilize them. The fact that the data describing a particular drill-through (the drill-through metadata) might be distributed over several sources meant that it was also difficult to integrate it with

such third-party applications. Additionally, each of these applications likely has different data format requirements.

Summary of the Invention

According to one aspect of the invention, there is provided a

The present invention, particularly when incorporated in a drill-through modeling tool, solves or alleviates the above-mentioned problems, as well as providing several other advantages as will be made clear in the following description.

A drill-through modeling tool (DMT) using the method and system of the invention provides for all the elements affecting drill-through behavior (the drill-through metadata) to be aggregated at a single point, thereby allowing administration to be simplified, and also permitting easier understanding and integration with third-party tools.

Such a DMT also provides graphical displays of drill-through paths for a DMT user or modeller. These displays show the parameters and dependencies of each drill-through path and allow users to obtain a quick overview of the drill-through network and further, they allow the tool user or modeller to confirm drill-through dependencies at a glance. Drill-through objects may thus be manipulated and maintained in a graphical manner.

A further understanding of other aspects, features and advantages of the present invention will be realized by reference to the following description, appended claims, and accompanying drawings.

Brief Description of the Drawings

A preferred embodiment(s) of the present invention will be described with reference to the accompanying drawings, in which:

Figure 1 shows the computer and related systems and connections within which the invention may be practiced;

Figures 2a and 2b show two aspects of the relationships between the data, the data modeling tool, the drill-through metadata, and the report generating applications; and

Figure 3 shows an example of drill-through network in which the invention may be used.

Detailed description of the Preferred Embodiment(s)

Typically, the invention is practiced within a network of computers and related equipment, an example of which is illustrated in Figure 1. The data are stored in a data warehouse comprising files servers 100, 102, 104 and storage devices 106, 108. The file servers 100, 102, 104 contain one or more software report generating applications, including modeling and administration tools using these data. The file servers 100, 102, 104 are accessed from workstations 110, 111, 112 over a network 120 which may be an intranet, Internet, or any other arbitrary network topology.

In one preferred embodiment, a Business Modeling Tool (BMT) framework is enhanced with a drill-through administration tool that allows the tool user or modeler to design a single drill-through solution comprising all of the metadata for drill-through sources and targets for an entire suite of business data analysis products, or in some cases a set of otherwise independent business data analysis products. The drill- through metadata is that information necessary to describe all of the required or identified drill-through objects and their paths. Other embodiments permit access to this information (the drill-through metadata) to be extended to external applications as well. Yet further embodiments permit the data accessed by the drill-through paths to be resident outside of the data warehouse, for example historical commodity prices residing on a well-known Hypertext markup language (HTML) server attached to the Internet.

In describing the invention, it is helpful to understand how the BMT and the drill-through service relate to the user of the data (who ultimately does the interpretation and analysis of the data) and to the report generating applications available, as well as to the data itself. The drill-through solution may be

considered as being implemented in two parts. The first part is the drill-through modeling tool that is used in performing the task of defining drill-through paths between various source and target objects. That is, metadata for a set of drill-throughs are defined and definitions of the drill-throughs are aggregated in a single point. The second part is the runtime environment that comprises those drill-through services and queries used by report generating applications, i.e. interfacing the meta-data to various data collections, such as data cubes or data-based reports, which are derived from different report generating applications. These parts are shown in Figure 2a where various report generating applications, 211, 212, and 213 make use of a runtime environment 220 to access metadata that defines valid drill-through path options using drill-through queries 225 on information stored as drill-through metadata 230. The drill-through metadata 230 is provided through a series of interactions 235 by the modeling environment 240. To reduce complexity the common sources of data for the modeling and runtime environments are not shown. Figure 2b shows in more detail the mechanism for accessing the drill-through metadata 270 from a typical report generating application 250, by making use of a drill-through service 260. When a drill-through is determined to be required by the application 250, under the control of a user (not shown) one or more queries through an API (application program interface) 255 are made of the drill-through service 260. The drill through service 260 reads through a further API 265 the drill-through metadata 270 in the course of responding to the one or more queries 255. Examples of queries that report generating applications 250 (such as ReportNet™ or Powerplay™ by Cognos Corporation) might issue through the API are:

- Generate a list of potential drill-through targets for a specified report (source).
- Generate a target report for a specified target and context information (where an example of the context information is a row of data from the report).

Figure 3 shows a representative sample of a drill-through network that might be displayed in a diagram view within the BMT. Referring to Figure 3, all

drill-through sources and targets are represented as nodes. Any node may be associated with any other node by means of a drill-through path, shown diagrammatically as an arrow between the relevant nodes. This particular example shows a number of reports generated by different report generating applications. The type 1 reports Corporate Customer List 310, Customer Mailing List 320, and Order Details 330, and the type 2 report Expenses 340 are distinguished in that they are typically derived from different report generating applications. These report generating applications may typically derive their data from different types of source (for example OLAP sources and relational database sources). In addition, there are two cubes, Corporate Sales 325, and Human Resources 335, derived from corporate data sources. One 'external' source of information is shown, Current Stock Prices 315, where the reference is made via an Internet Uniform Resource Locator (URL). A number of drill-through paths are identified, 312, 314, 316, 318, 322, 324, 326, 328 in which the arrowheads designate the direction of the drill-through. A list of drill-through targets is presented by the report generating application either using the drill-through path names or the target names. The drill-through paths made available to the user by the report generating application may have been identified by the tool user or modeller, or optionally by an automatic internal process. If there is more than one path to a drill-through target then it is the designer's responsibility to provide meaningful path names. Assigning these path names is optional as, for example, paths 322, 326, 328 have names assigned to them as follows: By Employee 322, By Sales staff 326 and By Sales manager 328. In cases where there are multiple paths to a single target and the paths do not have names assigned to them, the user of the report generating application, when presented with a drill-through target list dialog, would see duplicate entries, one for each path.

It should be noted that in some cases a drill-through path might be defined between two reports, such as 314, where the drill-through path extends from one report, Corporate Customer list 310, to another report, Customer Mailing list 320. In other cases, more than one drill-through path 326, 328 is defined between two nodes, such as between the Order Details report 330 and

the Expense report 340. There are no limits on connectivity of the drill-through paths, as long as they are logically correct - the selection of such paths is part of the drill-through designer's task.

Adding drill-through source and targets is similar to an import action. It should be realized that the details of the process of adding a drill-through object (source or target) are specific to the type of object.

The following are the steps performed in adding or importing a drill-through object:

1 - locate and open the object.

2 - determine what parameters could be used as outputs or could be part of the drill-through predicate.

3 - determine the parameters that could be used as inputs to the drill-through object. For example, report columns, dimension levels, or prompts.

4 - determine any application-specific information.

5 - create the actual drill-through object in the drill-through metadata.

The tool user or modeller is requested to provide the location of the object that is to be imported (such as a report) by creating a drill-through path. One mechanism to achieve this is by connecting the source and the target with a line on a graphical interface. A dialog box is then displayed which includes the source and target parameters in list boxes. The tool user or modeller can then match up the parameters by selecting equivalent pairs of parameters for the source and target from these list boxes. The editing of the drill-through paths may optionally employ a form of dialog or wizard as appropriate to the user's competence and desire to select appropriate parameters. The tool user or modeller may optionally add parameter-mapping functions, where appropriate, using similar list box tools or their equivalent.

In one preferred embodiment the drill-through modeling tool, after importing the object, automatically determines the possible drill-through paths by

examining the parameters of the imported object and attempting to find matching parameters with parameters of other drill-through objects. This list of potential drill-through paths is then presented to the tool user or modeller in a manner allowing selection of one or more desired paths.

In further preferred embodiments the "import" process attempts to determine the physical source on which a parameter is based. This matching process is carried out automatically and the matches presented as a list. The tool user or modeller then selects or deselects the matches from those presented. In these embodiments, if the source and target parameter names do not match exactly then other information regarding the parameters is used as a clue as to which source and target parameters match. For example a parameter may represent a report column and that column is based on a database column. The reference to that database column is saved and used as a 'hint' when trying to establish a mapping from a parameter from one drill-object source to target. As before, the tool user or modeller selects or deselects from a list to make a selection.

Typically, drill-through objects (source and targets) have their implementation-specific actions encapsulated within a dynamically shared library, within what are known as plug-ins. Alternatively, they may form part of a program library. Each plug-in is responsible for at least the following functions:

- The import actions, which is the act of determining the parameters for the drill-through object.
- Providing the actions and logic that the drill-through service needs to send either to perform the drill-through action or return the required data to the calling client of the drill-through service. When the drill-through service is queried and asked for a list of targets based on a drill-through predicate and its source, the drill-through server responds with a list of targets plus other associated information (such as a URL). The inclusion of a URL means that the client making the query could, based on the response, 'launch' the URL, thereby accessing the data contained in the referred-to location. The drill-through service also has actions that can be invoked by the client when it is

required to perform the action on behalf of the client. Typically, the drill-through service invokes the URL (for example by sending back a redirect HTML page to the client's browser). The plug-in specific to the drill-through target determines which action is performed. In some embodiments, the action queries one or more third party databases or applications (or more strictly the data or reports associated with or generated by the applications).

- A validation action that, when invoked by the modeling application, validates the existence of the target and that the parameter and other properties still apply/exist.

Other actions can also provided as required to perform any translations or reformatting of data between any two compatible applications.

The use of a plug-in or an equivalent library, means the metadata for each drill-through is stored in a single location, in a fixed and defined format, thereby simplifying the updating of such drill-throughs, and further simplifying the disclosure and publication of APIs and their related drill-through metadata for the use of third-party applications if necessary.

While various preferred embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. It will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined in the appended claims. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A method of providing a drill-through service between a plurality of drill-through objects (drill-through source and targets), the method comprising steps of:

(a) defining one or more drill-throughs, the definitions of the drill-throughs being agregated at a single place; and

(b) interfacing the drill-throughs to the drill-through objects in a run-time environment; and

(c) wherein the drill-throughs are independently administered and maintained.

2. A method of claim 1, wherein the drill-through objects includes data collections which are derived from different applications.

3. A method of claim 2, wherein the data collection includes data cubes and data-based report, which are derived from different report generating applications.

4. A method of claim 1, wherein the definition of a drill-through includes metadata therefor.

5. A database interface for providing a drill-through service between a plurality of drill-through objects (drill-through source and targets), the interface comprising:

(a) means for defining one or more drill-throughs, the definitions of the drill-throughs being agregated at a single place; and

(b) run-time environment means for interfacing said drill-throughs to the drill-through objects; and

(c) wherein the drill-throughs are independently administered and maintained.

6. A database interface of claim 5, wherein the drill-through objects includes data collections which are derived from different applications.

7. A database interface of claim 6, wherein the data collection includes data cubes and data-based report, which are derived from different report generating applications.

8. A database interface of claim 5, wherein the definition of a drill-through includes metadata therefor.

9. A drill-through administration method for use in a framework having a plurality of sources and targets, the sources and targets having potential drill-through relationships, the method comprising steps of:

(a) displaying the potential drill-through path sources and targets;

(b) accepting from the tool user or modeller those sources and targets for which a drill-through path is required;

(c) for each source for which a drill-through path is required;

(i) importing the source;

(ii) optionally determining automatically the possible drill-through paths for the required sources and targets;

(iii) permitting the tool user or modeller to select one or more drill-through paths;

(iv) allowing the tool user or modeller to edit the selected drill-through paths to select appropriate parameters;

(v) allowing the tool user or modeller to edit the selected drill-through paths to add parameter mapping functions; and

(d) encapsulating the drill-through paths in a program library.

10. The drill-through administration method of claim 9, wherein the step of accepting from the tool user or modeller those sources and targets for which a drill-through is required uses a graphical user interface whereon the user draws lines connecting nodes representing the sources and targets.

11. The drill-through administration method of claim 9, wherein the step of optionally determining automatically the possible drill-through paths for the required sources and targets comprises the steps of:

(a) comparing the source and target parameter names;

(b) if the source and target parameter names match then establishing a mapping between the source and target parameters;

(c) if the source and target parameter names do not match then perform the steps of:

(i) searching for other information regarding the parameters which matches and establishing a preliminary mapping between those sources and targets;

(ii) presenting the tool user or modeller with a list of preliminary mappings from which to make a selection; and

(iii) adding the selected preliminary mappings to the list of mappings established by matching parameter names.

12. The drill-through administration method of claim 9, wherein the program library is a dynamically shared library.

13. The drill-through administration method of claim 9, wherein the program library is a plug-in.

14. The drill-through administration method of claim 9, wherein the source comprises one or more databases or applications optionally provided by a third party.

15. A computer-based drill-through administration tool for use by a tool user or modeller within a computer-based Business Modeling Tool with a framework composed of source and targets having potential drill-through relationships, the drill-through administration tool consisting of:

(a) means for displaying the potential drill-through path sources and targets;

(b) means for accepting from a tool user or modeller those sources and targets for which a drill-through path is required;

(c) for each source for which a drill-through path is required;

(i) means for importing the source;

(ii) optional means for determining automatically the possible drill-through paths for the required sources and targets;

(iii) means for permitting the user to select one or more drill-through paths;

(iv) means for allowing the tool user or modeller to edit the selected drill-through paths to select appropriate parameters; and

(v) optional means for allowing the tool user or modeller to edit the selected drill-through paths to add parameter mapping functions; and

(vi) means for encapsulating the drill-through paths in a program library.

16. The drill-through administration tool of claim 15, wherein the means for accepting from the tool user or modeller those sources and targets for which a drill-through is required uses a graphical user interface whereon the tool user or modeller draws lines connecting nodes representing the sources and targets.

17. The drill-through administration tool of claim 15, wherein the means for optionally determining automatically the possible drill-through paths for the required sources and targets consists of:

(a) means for comparing the source and target parameter names;

(b) if the source and target parameter names match then providing means for establishing a mapping between the source and target parameters;

(c) if the source and target parameter names do not match then providing:

(i) means for searching for other information regarding the parameters which matches and establishing a preliminary mapping between those sources and targets;

(ii) means for presenting the tool user or modeller with a list of preliminary mappings from which to make a selection; and

(iii) means for adding the selected preliminary mappings to the list of mappings established by matching parameter names.

18. The drill-through administration tool of claim 15, wherein the program library is a dynamically shared library.

19. The drill-through administration tool of claim 15, wherein the program library is a plug-in.

20. The drill-through administration tool of claim 15, wherein the source comprises one or more databases or applications optionally provided by a third party.

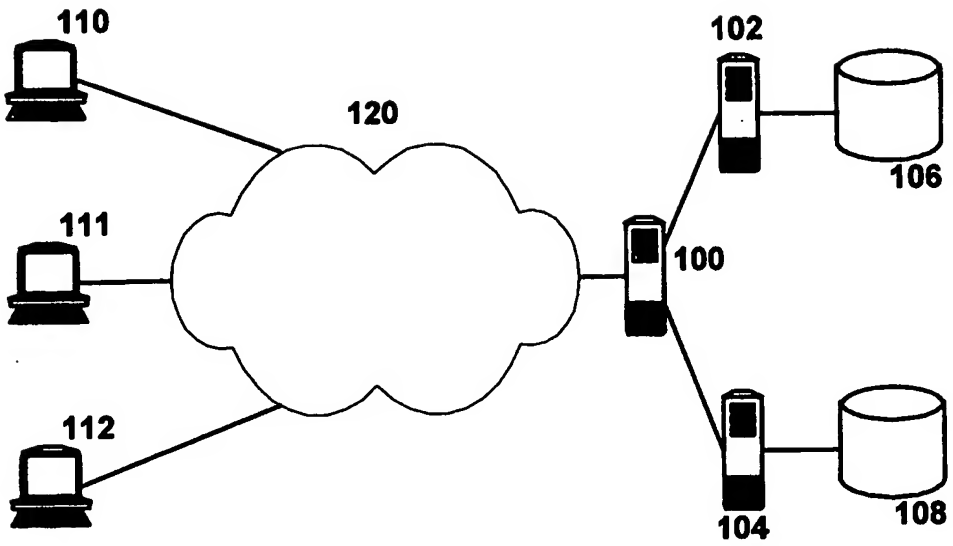

FIG. 1

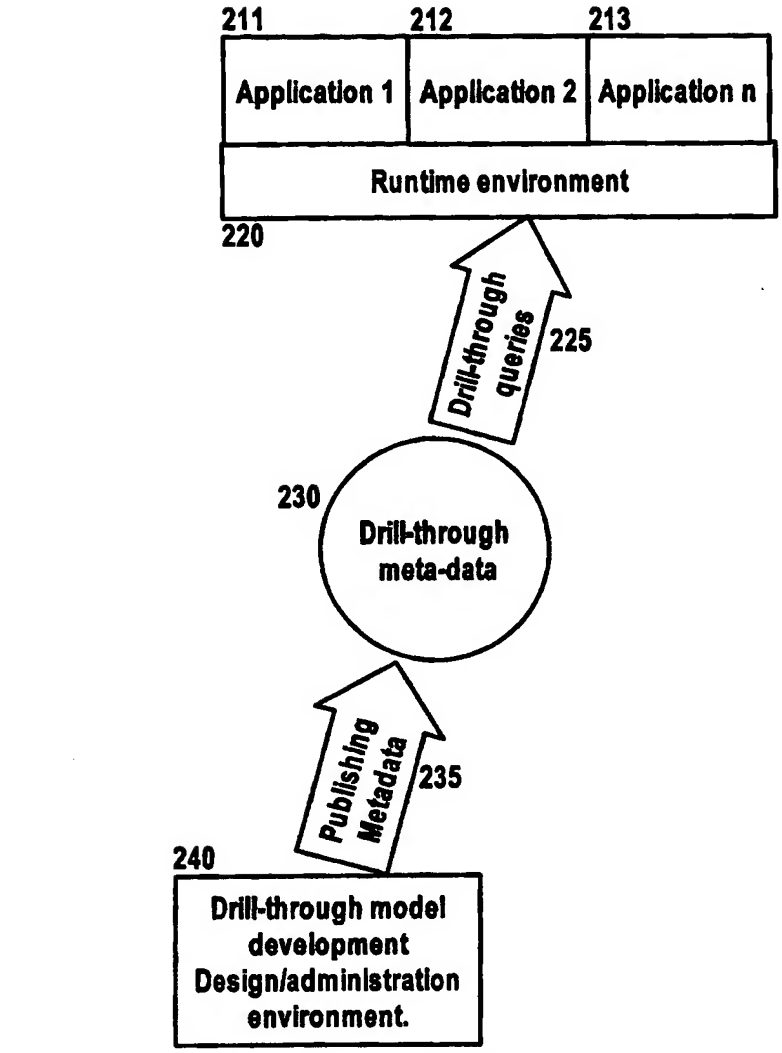
211
212
213
Application 1
Application 2
Application n
Runtime environment
220
Drill-through queries
225
230
Drill-through meta-data
Publishing Metadata
235
240
Drill-through model development Design/administration environment.

FIG. 2a

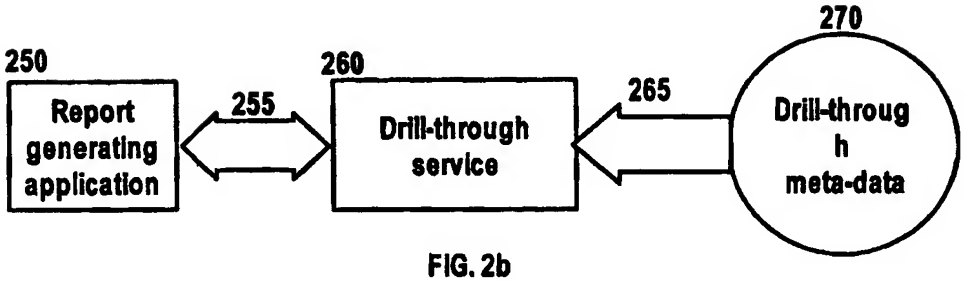
270
250
260
265
Report generating application
255
Drill-through service
Drill-throug h meta-data

FIG. 2b

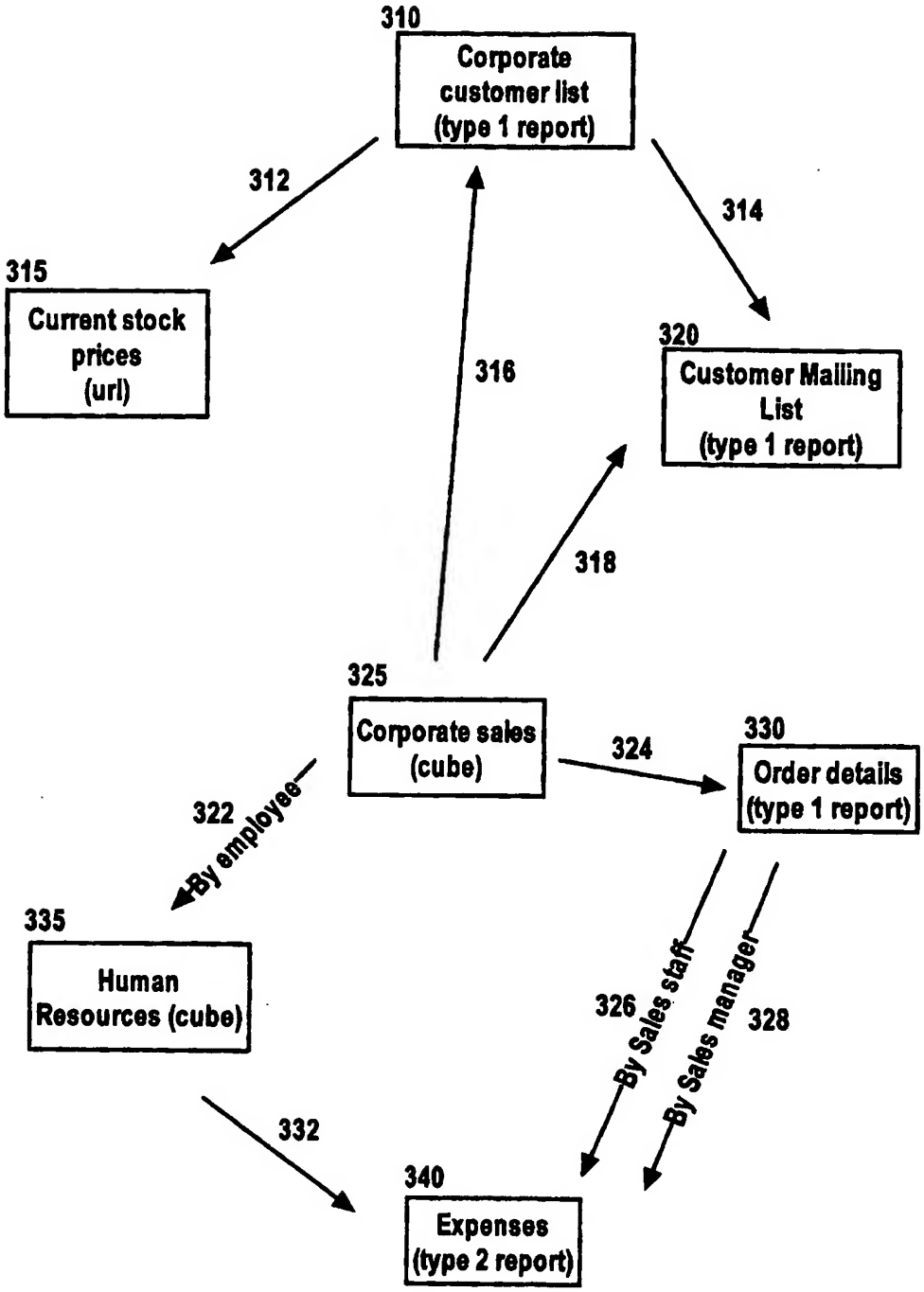
310
Corporate customer list (type 1 report)
312
314
315
Current stock prices (url)
316
320
Customer Mailing List (type 1 report)
318
325
Corporate sales (cube)
324
330
Order details (type 1 report)
322
By employee
335
Human Resources (cube)
326
By Sales staff
328
By Sales manager
332
340
Expenses (type 2 report)

FIG. 3

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.
1300 I Street, N.W.
Washington, D.C. 20005

SERIAL NO: 10/624,489

DOCKET NO: 08005.0009